NAS/NCET

Run Book

Document Category: RB

Version: 1.0.0

Status: Complete

Last Updated: 2024-05-09

# Table of Contents

[1 Table of Contents 2](#_Toc166136178)

[2 Introduction 3](#_Toc166136179)

[2.1 Purpose 3](#_Toc166136180)

[2.2 Scope 3](#_Toc166136181)

[2.3 Target Audience 4](#_Toc166136182)

[2.4 Related documents 4](#_Toc166136183)

[2.5 Glossary 4](#_Toc166136184)

[3 Installation and Setup 5](#_Toc166136185)

[3.1 Installation 5](#_Toc166136186)

[3.1.1 Pre-requisites 5](#_Toc166136187)

[3.1.2 Migration/Installation Steps 5](#_Toc166136188)

[3.2 Setup & Configurations 12](#_Toc166136189)

[3.2.1 AppPool Configuration 12](#_Toc166136190)

[3.2.2 Database Configuration 15](#_Toc166136191)

[4 Operations Procedure 19](#_Toc166136192)

[4.1 Basic Operations 19](#_Toc166136193)

[4.1.1 Start-up and Initial Validation 19](#_Toc166136194)

[4.1.2 Shutdown 19](#_Toc166136195)

[4.1.3 Restart 21](#_Toc166136196)

[5 Monitoring 22](#_Toc166136197)

[5.1 Determine that the system is functioning properly 22](#_Toc166136198)

[5.2 Determine that the system is performing properly 22](#_Toc166136199)

[Approvals 24](#_Toc166136200)

[History 25](#_Toc166136201)

# Introduction

## Purpose

The purpose of this document is to provide install and operational guidelines for the Nursing Clinical Evaluation Tool and will specifically cover the following topics:

* Installation
* Environment preparation
* Install and post install verification
* Configuration & setup
* Operations Procedures
* Startup/shutdown/restart
* Anything that the operators might need to do to support the users
* Operational Monitoring

## Scope

This document's scope is to provide detailed instructions on deploying the application to the target server. The system must be deployed to the server, followed by creating an app pool for the system and database in the target environment (ex. CSDEV/CSTEST). Additionally, the system must be linked with AMS in the proper environment. AMS handles the login functionality of the application. See 3.1 for detailed step-by-step instructions on the installation.

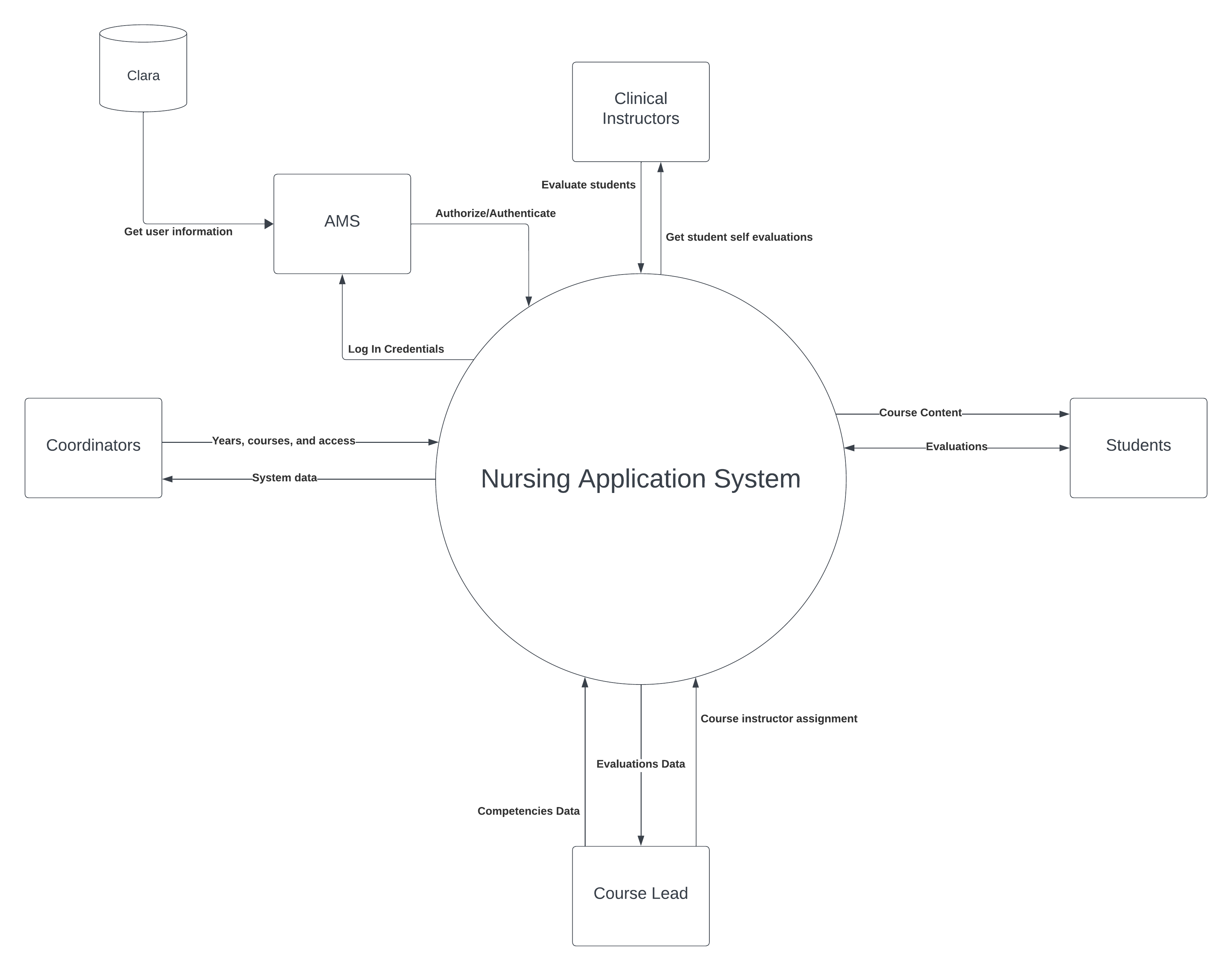
****

Figure 1: NCET Context Diagram

To better understand the system this diagram shows how the various components of the application interact with each other. Further explanation on the diagram can be found in NCET-ADS.

## Target Audience

This document is intended for use as a reference by infrastructure engineers and system operators responsible for the installation and subsequent operational support of the Nursing Clinical Evaluation Tool and related systems. This document is not intended for system users and does not intend to explain its design.

## Related documents

These documents contain information related to the information in this document.

| Document Short Name Reference | Document Title | Version (Optional) |
| --- | --- | --- |
| NCET-ADS | Architecture Design Standards |  |
| NCET-SD | System Documentation | 1.0.0 |

Table 1‑1 Related Documents

## Glossary

Acronyms mentioned throughout the document are described below.

| Term/Acronym | Description |
| --- | --- |
| NCET | Nursing Clinical Evaluation Tool (Front facing application name) |
| NAS | Nursing Application System (Internal system name) |
| CSDEV | Heritage IIS server for development  Server which application will be deployed and for development |
| CSTEST | Heritage IIS server for testing  Server which application will be deployed for testing |
| AMS | Application Management System. Heritage’s Auth application. |
| Rider | JetBrains IDE for C# and .NET development. |
| SSMS | SQL Server Management Studio. |

Table 1‑2 Glossary

# Installation and Setup

## Installation

### Pre-requisites

Before installing the NAS/NCET, please make sure the following environment requirements are met.

#### Hardware Requirements (Database Server):

* Software and hardware architecture of 64 bits
* Intel Xeon Gold 6142 CPU @ 2.60GHz (2 Processors)
* 1Gb of Ram
* 170Mb of Disk Space Per instance
* Network speed: 5Mbps (can move to 10Mbps when demand grows)

#### Hardware Requirements (Application Server):

* Software and hardware architecture of 64 bits
* Intel Xeon Gold 6142 CPU @ 2.60GHz (2 Processors)
* 1Gb of Ram
* 500Mb of Disk Space Per instance
* Network speed: 5Mbps (can move to 10Mbps when demand grows)

#### Software Requirements (Database Server):

* Windows Server 2019 standard
* Monitoring software (if required)
* Microsoft SQL Server 2018

#### Software Requirements (Application Server):

* Windows Server 2019 standard
* IIS Version 10
* Monitoring software (if required)
  + No custom or specific monitoring system is required.
* .NET Core Version 6

Note: The procurement and installation process of the required third-party hardware & software components mentioned above is outside the scope of this run book and is not described.

Source: Specs taken from the Heritage Co-op Management System (HCMS) runbook document

### Migration/Installation Steps

This documentation will presume the usage of Rider. It will mostly work the same in Visual Studio, but Rider is the recommended IDE for this project.

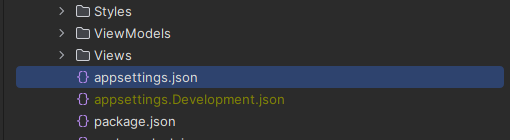
#### Checkout branch to deploy

This will likely be the “main” branch; this can be done through the Git CLI or via the UI in Rider.

Remember not to commit or push any of the changes made during the deployment process.

#### Change “appsettings.json” to target the correct database

This file is located at NAS/appsettings.json.



The server URL will be something like “cssql.cegep-heritage.qc.ca”.

The database name should be something like “NCET\_<developer initials>”, for me that’s NCET\_SCB (not my initials, but close enough).

*This database doesn’t need to exist yet, just use the name you’re going to use. In fact, if it does already exist you should probably delete it.*



#### Create your database via deployment script

*Complete this section* ***or*** *the next section.*

You can acquire the deployment script from the Git repository on Azure DevOps.

There may be an option for seedless and seeded. This determines whether test data is included in the database by default. Use **seedless** for production deployments.

You must login to the database server that you’re already targeting with the connection string. This can be done via SSMS, Rider, or DataGrip (whichever you’re more comfortable with).

Create a new query and copy the deployment script into it.

You’ll need to replace the database name references in the script with your desired name.

Run the script and your database will now be live.

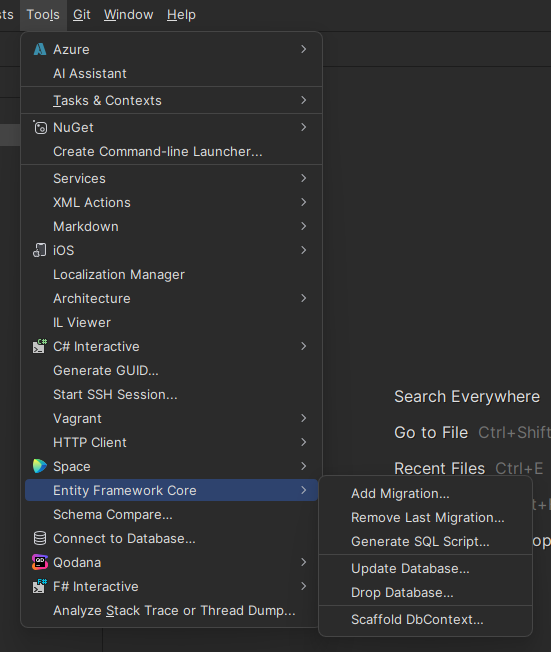
#### Create your database via migrations

*Complete this section* ***or*** *the previous section.*

You deploy via migrations easily via the interface in JetBrains Rider. This **requires** the Entity Framework Core plugin to be installed. If the plugin is not installed you can do this through settings.

**Note**: Migrations do have seeded data, if you require no seeded data, it will be easier to use the deployment script.

If the project doesn’t currently have migrations, you’ll need to generate them.



Update the database via the tool menu twice, first with migration 0, then with migration \_initial. Uncheck “Use the default connection of the startup project” to ensure you use the correction connection string.

This should take no more than a minute or so.

Since AMS is not working now, the next step bypasses the authorization and allows access into the system. To do so, replace the Login Task inside AccountSession.cs with the following:  
  
 public async Task Login(Credentials credentials) {

try {

// Entered valid username+password combination

// TODO: re-enable auth

var authenticated = true;//await Authenticate(credentials);

if (!authenticated) throw new AuthenticationException("Incorrect username or password");

// TODO: re-enable auth Part 2

// Gather user authorization / information

UserBLL userBll = null!; // await Authorize(credentials.Username);

switch (credentials.Username)

{

// TODO: Temporary fake user to get in

case "NC":

userBll = new UserBLL

{

UserId = 1789,

Username = "Coordinator",

FirstName = "Coor",

LastName = "Dinator",

RoleList = new RoleBLL[]

{

new RoleBLL

{

Code = "NC",

Description = "temp",

RoleId = 1

}

}

};

break;

case "NL":

userBll = new UserBLL

{

UserId = 1482,

Username = "Leads",

FirstName = "Lead",

LastName = "Er",

RoleList = new RoleBLL[]

{

new RoleBLL

{

Code = "NL",

Description = "temp",

RoleId = 4

}

}

};

break;

case "NI":

userBll = new UserBLL

{

UserId = 1234,

Username = "Instructor",

FirstName = "Instru",

LastName = "Ctor",

RoleList = new RoleBLL[]

{

new RoleBLL

{

Code = "NI",

Description = "temp",

RoleId = 3

}

}

};

break;

case "NS":

userBll = new UserBLL

{

UserId = 1239,

Username = "Student",

FirstName = "Stu",

LastName = "Dent",

RoleList = new RoleBLL[]

{

new RoleBLL

{

Code = "NS",

Description = "temp",

RoleId = 2

}

}

};

break;

}

// Setup claims

var processedRoles = await SetupClaims(userBll);

// Sync user data with AMS

await \_userRepository.SyncUserData(userBll.UserId, userBll.FirstName, userBll.LastName, processedRoles.ToArray());

}

catch (EndpointNotFoundException) {

// Likely means the project is being run outside the school network

throw new AuthenticationException(

"Auth server could not be reached, if this issue persists contact an administrator");

}

}

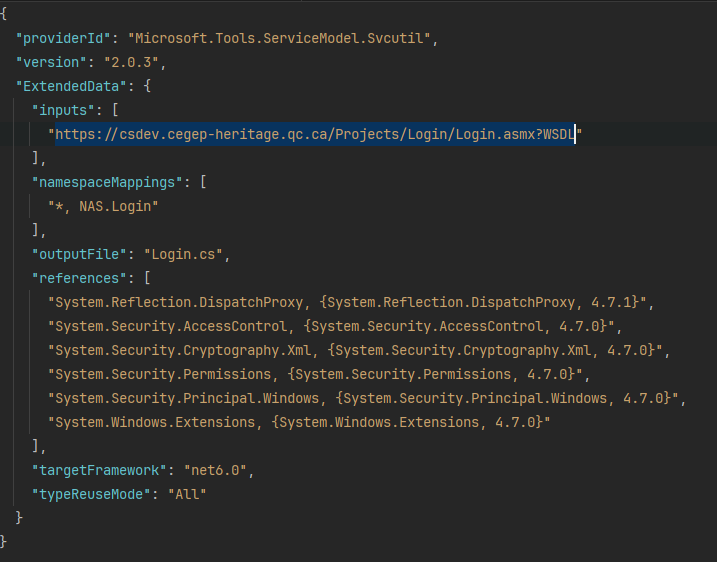
This will allow you to login as a coordinator, student, lead and instructor with the corresponding usernames NC, NS, NL, NI. The password does not matter. The rest of the document has been commented out due to the current state of AMS.

For an example on the initial setup, look at the Nursing\_F24\_Dev branch at the following link:

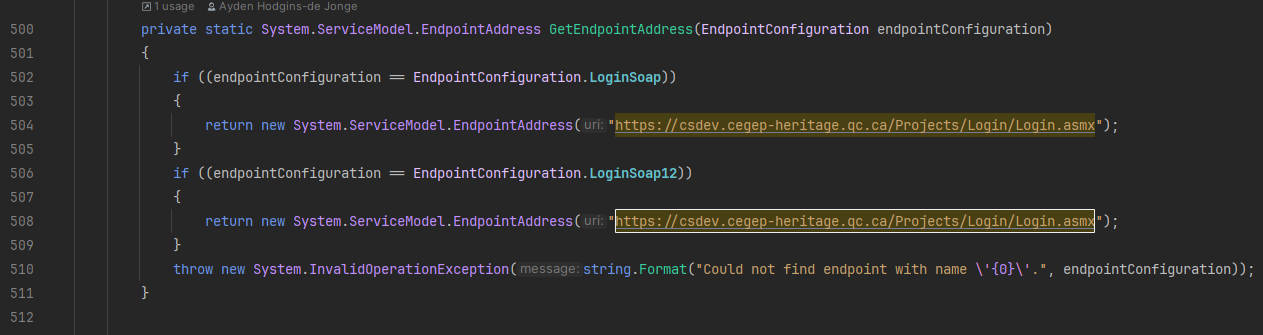
<https://csazure.cegep-heritage.qc.ca:8080/F2023-DevProject/Project%203%20-%20Nursing%20App/_git/Project%203%20-%20Nursing%20App>

#### *Update AMS ConnectedService*

*In NAS/Services/Login/ConnectedService.json adjust the URL to point to the correct AMS installation for your server.*

**

*You must also update the URLs in Login.cs, this file is in the same directory as ConnectedService.json.*

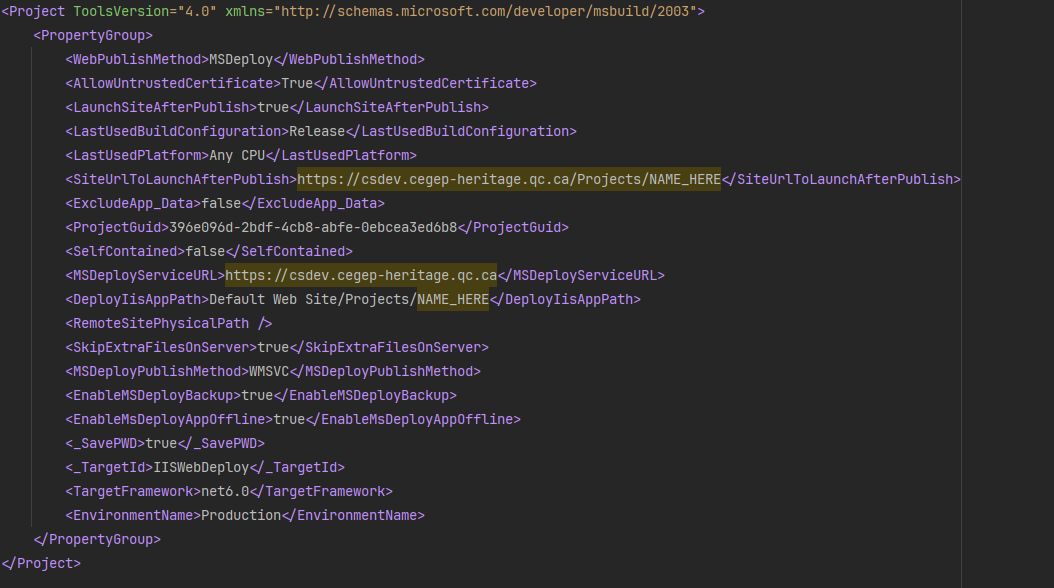
**

#### *Update “Remote.pubxml” with correct server*

*You need to modify the publish profile located at NAS/Properties/PublishProfiles/Remote.pubxml.*

*You need to specify the server you’re publishing to, and where it is located.*

*Modify the highlighted fields appropriately, generally, where NAME\_HERE is entered we’d use the same name as the database.*

**

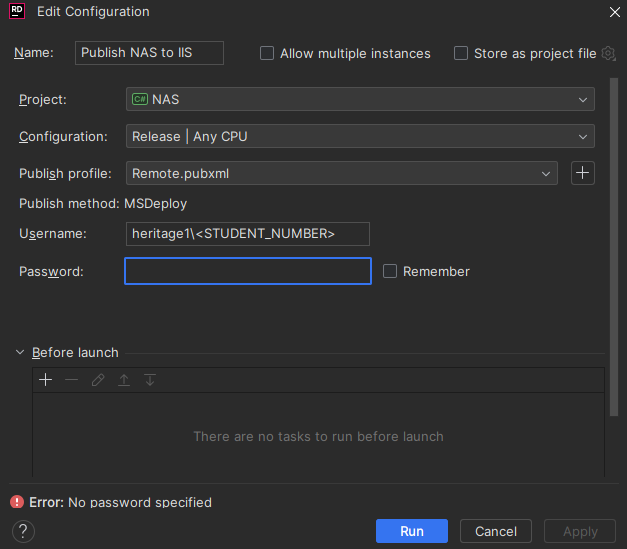
#### *Run the publish configuration*

*This might be called something like “Publish to IIS” or “NAS: Remote”.*

*The dropdown to select this configuration will be in the upper right corner of Rider.*

**

*Upon running it, you will need to enter your student number and password to be able to publish to CSDEV/CSTEST.*

**

*If you’re lucky then you may already be able to access the deployed version… it may even have access to the database. However, don’t be fooled, this setup is precarious and needs to be adjusted.*

*In the next section, you’ll learn how to set up the application pool properly.*

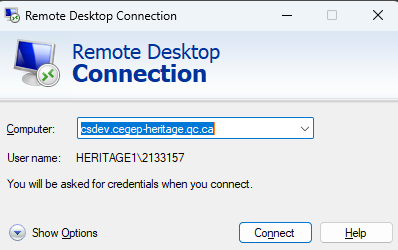
## *Setup & Configurations*

*In this section, we’ll configure the application (on the remote server) to be able to run in an isolated environment and access its database.*

### *AppPool Configuration*

#### *Connect to remote server desktop environment*

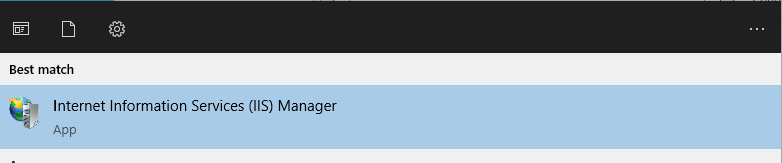
*You will need to connect to a remote machine to access IIS to set up the AppPool.*

**

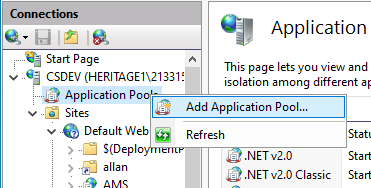
*This may take a few minutes to get connected, also only two students can be connected at once.*

#### *Setup the AppPool for your deployed version*

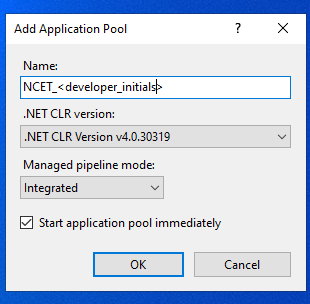
*You’ll need to open IIS.*

**

*Inside IIS you’ll navigate to Application Pools within the server, in this case CSDEV.*

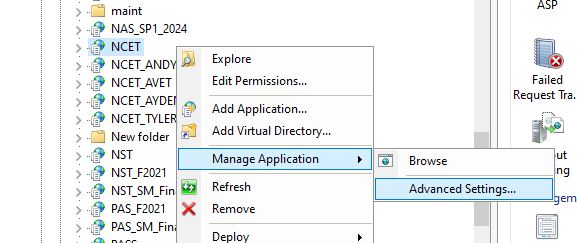
**

*Now you can create a new AppPool. Generally, we use the same name that we used for our database and application for this.*

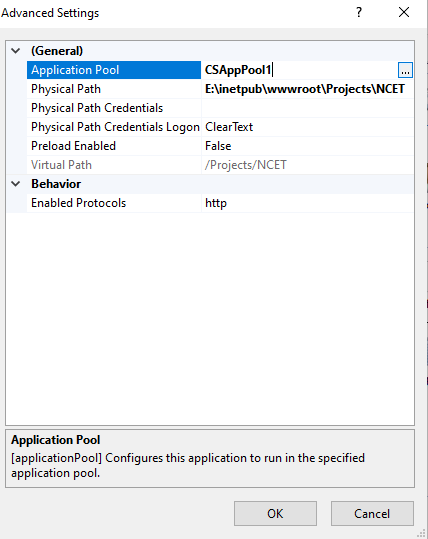
**

*Now you need to find your deployed application. In this case, it is located at CSDEV/Sites/Default Web Site/NCET\_<developer\_initials>.*

*You will right-click on it and navigate to Advanced Settings.*

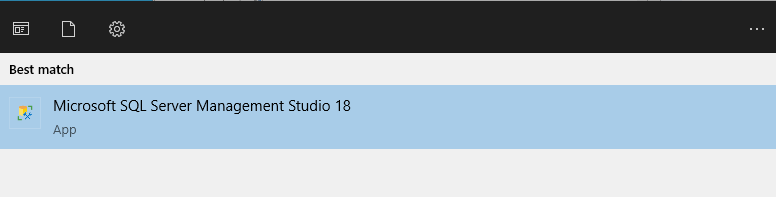
**

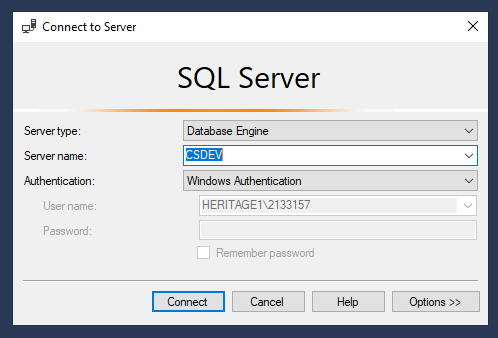
*From Advanced Settings you will change the AppPool to your freshly created one. Pressing ok will save this change.*

**

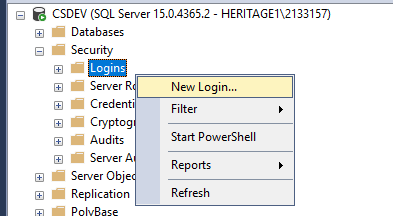
### *Database Configuration*

*Open SSMS and login to the server your database is located on.*

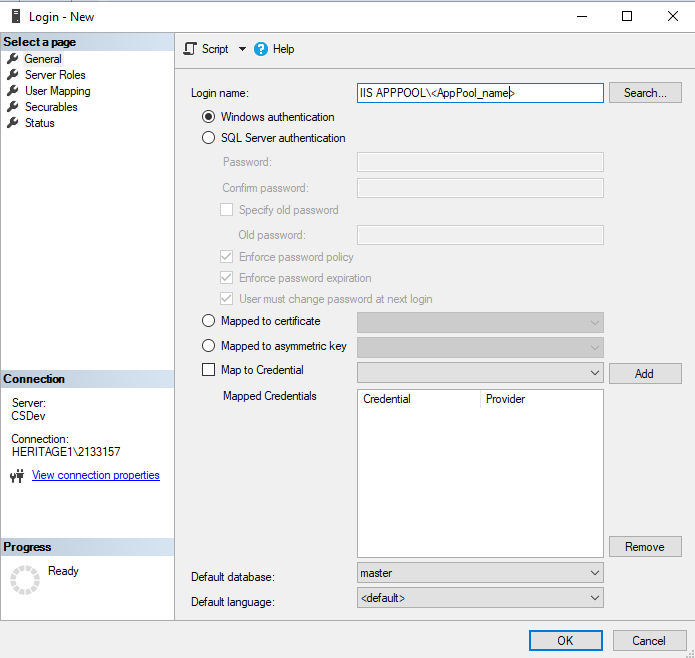
**

**

*You will create a new login, this gives the AppPool, and by extension the application itself access to its database.*

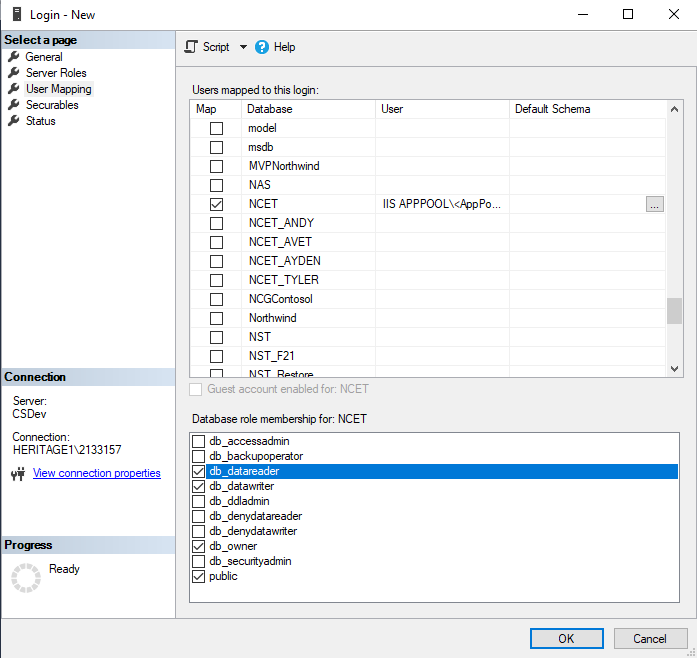
**

*The first step is to specify the login name, which will follow the format “IIS APPPOOL\<AppPool\_name>”, for example “IIS APPPOOL\NCET\_AYDEN”.*

**

*Next, you will navigate to “User Mapping” via “Select a page” on the left-hand side of the window.*

*You must find your database in the list, and enable the permissions “db\_datareader”, “db\_datawriter”, and “db\_owner”.*

**

*Huzzah! After clicking “Ok” your deployment should be fully functional.*

# *Operations Procedure*

## *Basic Operations*

### *Start-up and Initial Validation*

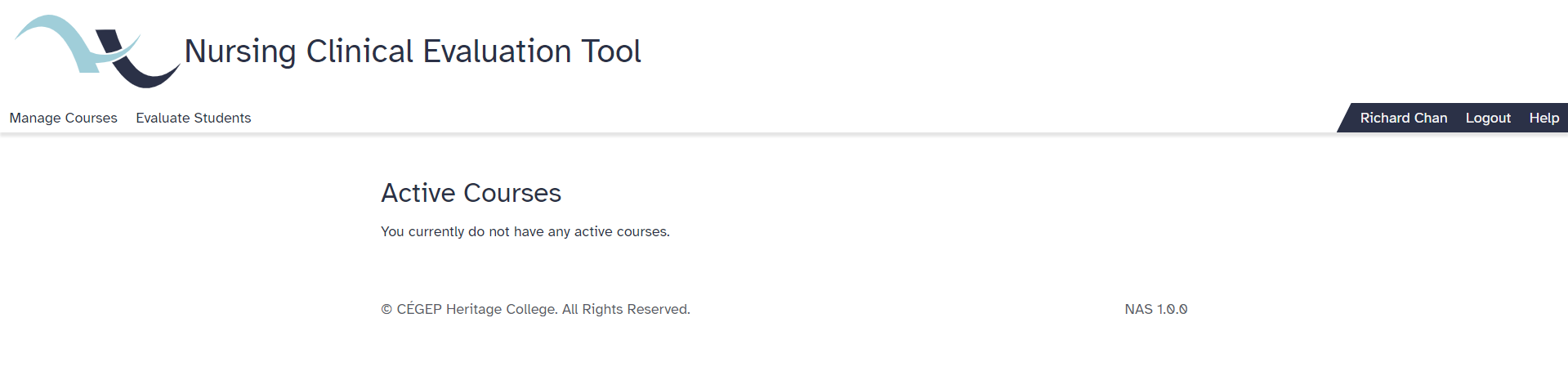
*Complete the steps required to deploy the system in the previous sections (****section 3.1****).*

*Using your preferred browser, access the deployed project via the server URL.*

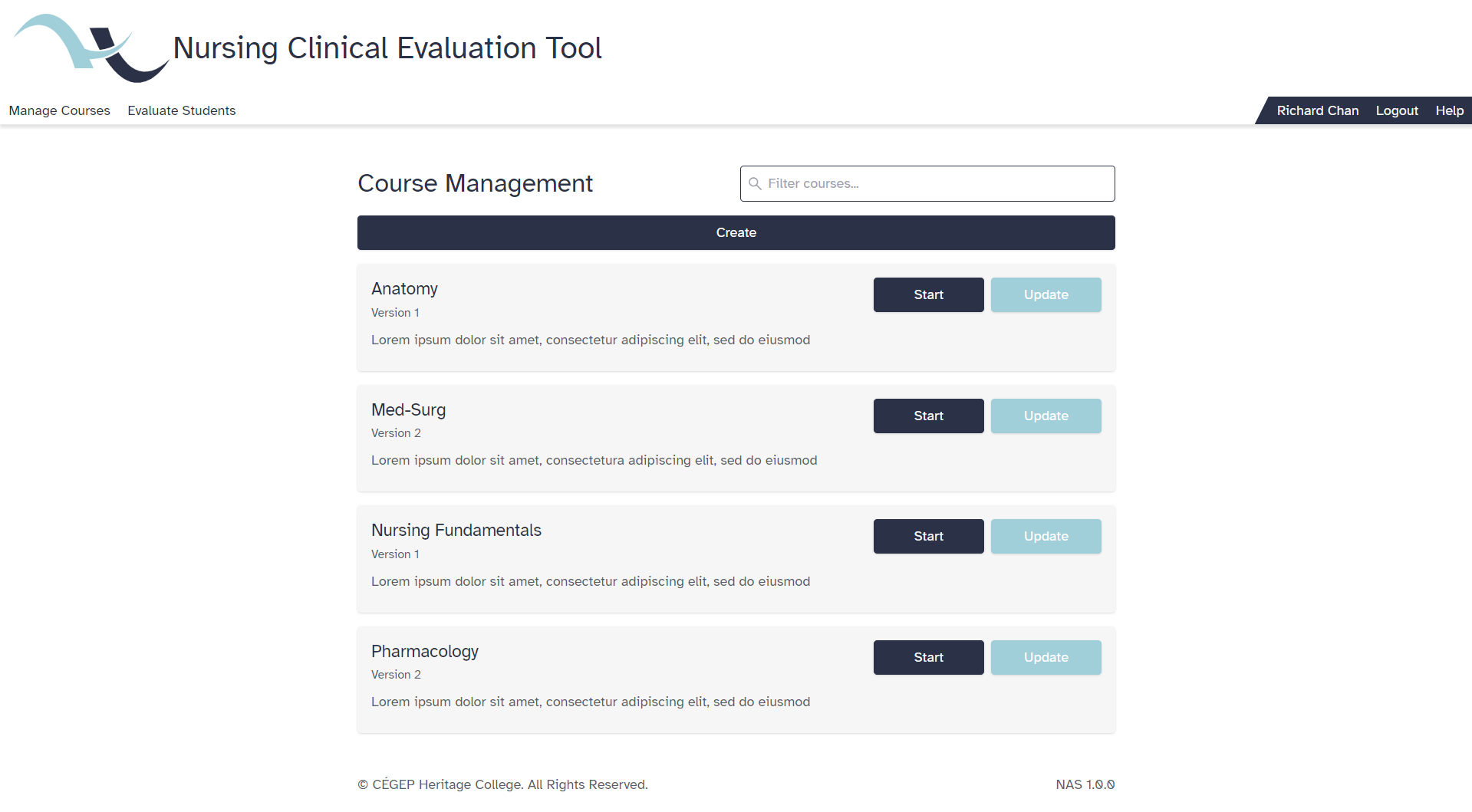
*Attempt to log in as a nursing coordinator.*

*Ensure the Manage Courses and Evaluate Students pages are working.*

*Evaluate Students page:*

**

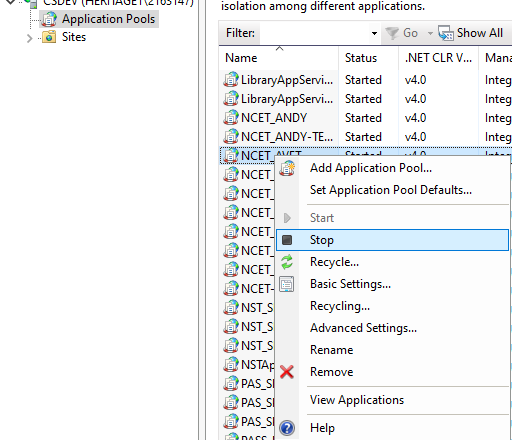
*Manage Courses page:*

**

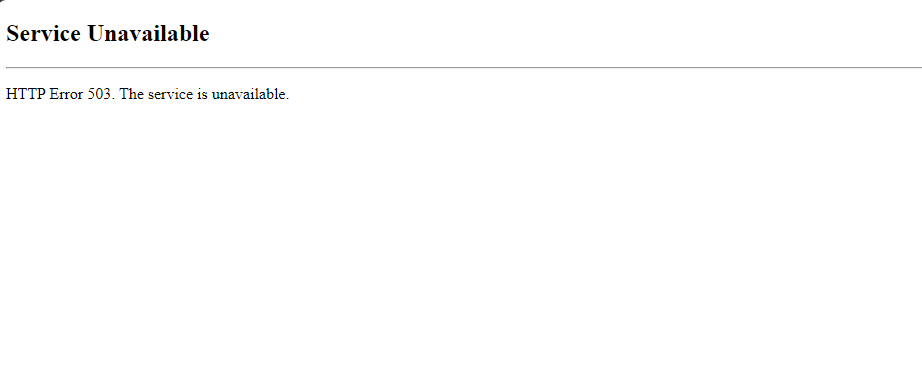
### *Shutdown*

*Remotely connect to the server IIS Manager\Application Pools.*

*Right-click and stop the App Pool in use for the system to shut it down.*

**

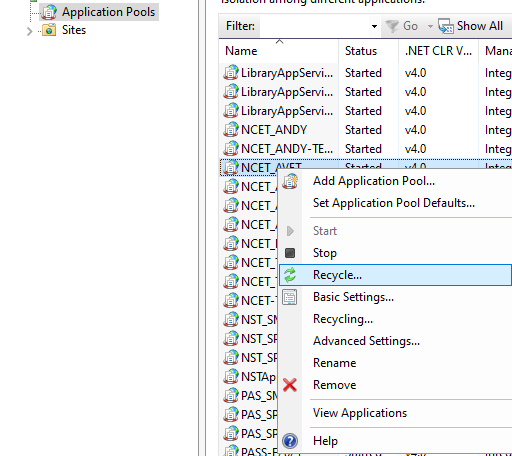
*Verify that the system has stopped successfully by accessing the project URL and validating that the service is unavailable.*

**

### *Restart*

*Remotely connect to the server IIS Manager\Application Pools.*

*Right-click and “Recycle” the App Pool in use for the system to restart it. You can also “Stop” the app pool and “Start” it again.*

**

*Verify that the system is running successfully by accessing the project URL.*

# *Monitoring*

## *Determine that the system is functioning properly*

*We will determine that the system is up and running by checking a few key things are working as expected.*

1. *Login to the system as a student authorized account (NS in AMS).*
   1. *You should see a “My Courses” and “Self Enrol” tab in the navigation bar.*
2. *Go to “Help”, located on the right of the navigation bar.*
   1. *You should only see one item underneath the courses section.*
3. *Logout*
4. *Login to the system as a lead authorized account (NL in AMS).*
   1. *You should see a “Manage Courses” and “Evaluate Students” tab in the navigation bar.*
5. *Go to “Manage Courses”.*
   1. *You should* ***NOT*** *see a create button on this page.*
6. *Logout*
7. *Login to the system as a coordinator authorized account (NC in AMS).*
8. *Go to “Manage Courses”.*
   1. *You should see a create button on this page.*

*This will ensure that some basic aspects of auth & auth are working. This is not extensive testing but*

## *Determine that the system is performing properly*

*To determine that the system is performing properly you must perform a basic course run creation.*

1. *Login as four separate student authorized accounts (NS in AMS).*
   1. *This is to ensure the system has students saved to its database.*
2. *Login as a coordinator authorized account (NC in AMS).*
3. *Via “Manage Courses” create a course.*
   1. *This course should have three competencies.*
   2. *Each competency should have 6 criteria.*
4. *Now, again via “Manage Courses”, start a new course run of the freshly created course.*
5. *Enter the course run via the “Manage” button.*
6. *Click “Add instructors/students to run”.*
7. *Now select four students and save your changes.*
   1. *This should take less than 2 seconds (much less in fact).*

*This will ensure that the system is handling its connection to the database fast enough.*

Approvals

This document has been read and approved by the following people, responsible for its implementation. Approval is indicated by an email showing approval. Those approving below indicate that the contents of this document are correct and complete and agree to their implementation:

| Title | Name | Approval |
| --- | --- | --- |
| Developer | Tyler Saikaley-Theriault | Approved |
| Developer | Andy Atabe | Approved |
| Developer | Ayden Hodgins-de Jonge | Approved |
|  |  |  |

History

| Version | Status | Date | Author | Reason for changes |
| --- | --- | --- | --- | --- |
| 1.0.0 | Complete | 2024-05-09 | Avetik Hakobyan  Ayden Hodgins-de Jonge  Andy Atabe  Tyler Saikaley-Theriault | Initial runbook |
| 1.1.0 | Complete | 2024-09-19 | Sebastian Canales Burke | AMS is broken |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |